International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

FPGA Interpolators With Reconfigurable and Multifrequency Approximate Computing using HEVC Fractional Pixel

Suma Gurulingaiah Charantimath[1], Ravi S M[2]

[1]Assistant Professor, Electronics and Communication Department, PDIT College, Hospet, Karnataka, India

[2]Assistant professor, Mechanical Department, SIT College, Mangalore, Karnataka, India

Abstract: ***Applicable in different fields and markets, low energy high efficiency video coding (HEVC) codecs and their constituting elements have been heavily studied. Fractional pixel interpolation is one of its most costly blocks. In this work, a field programmable gate array implementation of HEVC fractional pixel interpolation, outperforming literature solutions, is proposed. Approximate computing, in conjunction with hardware reconfiguration, guarantees a tunable interpolation system offering energy versus quality tradeoff to further reduce energy.***

Keywords: Embedded applications, field programmable gate array (FPGA), FIR filters, low power architectures, low power design, reconfigurable computing, runtime reconfiguration, signal processing

1. Introduction

The high efficiency video coding (HEVC) video compression standard, defined in 2013 by ITU-T video coding experts group jointly with the ISO/IEC moving picture experts group, is one of the latest released codecs. It addresses modern embedded video systems: high performance and high compression ratio make it possible to process and exchange video in real-time. HEVC provides a gain of up to 50% in terms of subjective video quality with respect to previous standards [1]. With the recent progress of system-on-chips (SoCs), embedded video decoders for the HEVC standard are now a reality. However, providing power efficient designs to cope with the compelling demand for long battery life is still a challenging task.

The approximate computing paradigm is a well suited solution to reduce energy consumption by exploring the tradeoff between energy and quality of the system output. Approximate computing provides three degrees of freedom by acting on data, hardware, and/or computation [2]. Approximating the data consists in reducing the quality of the application in a controlled way by processing either less up-to-date data (temporal decimation), less input data (spatial decimation [3]), less accurate data (word-length optimization [4]), or even corrupted data. In the hardware degree of freedom, the exactness of computation can be relaxed by using inexact operators [5] or voltage over scaling. The third degree of freedom, corresponding to computation and algorithm modifications, is under consideration in this letter. It aims at approximating the processing to decrease the computational complexity [6]. To be effective, algorithm-level approximations must target computation intensive blocks. HEVC computation is dominated by motion compensation [6], [7]. This latter, implemented with fractional pixel interpolation filters, represents between 62% and 80% of a decoder complexity. Nogues *et al.* [6] proposed to approximate HEVC fractional pixel interpolation filters and show that this approximation allows for saving up to 28% of software decoder energy, at the cost of minimal decoded video quality degradation. These results strongly motivate further studies on adaptable HEVC decoders, tuning image quality at runtime basing on an energy budget. Software implementations offer the flexibility required by modern applications but at the expense of high energy consumption compared to hard-wired solutions. The specialization of computation resources through the design of hardware accelerators for the most intensive parts is known to be the right approach to improve energy efficiency. Gomez-Pulido *et al.* [8] suggested that field programmable gate array (FPGA) implementations may be orders of magnitude more power efficient than software ones. To the best of our knowledge, the concept of approximate filters for HEVC decoders, presented in [6], has not been exploited in hardware prior to a previous preliminary study [9]. In this letter, a reconfigurable architecture is proposed for HEVC fractional pixel interpolation filters, exploiting the concept of algorithmic level approximation proposed in [6]. An energy-aware FPGA implementation of this architecture is carried-out and evaluated. The distinguishing feature of the proposed approach is the adoption of reconfigurable approximate computing: coarse-grained reconfiguration [10], [11] provides a tradeoff between a controlled image quality degradation and substantial energy saving. In [9], the same approach has been used on a smaller example, limited to a single channel interpolator. In this work, it:

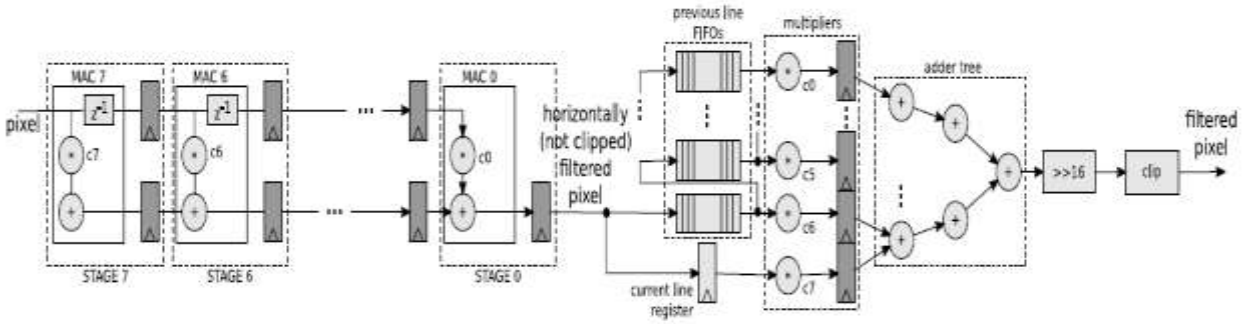

Figure 1: 2-D legacy 8 tap interpolation hardware architectures

1) Introduce architectural optimizations that bring the work introduced in [9] to the full scale, multichannel, and reproducible, implementation presented in Section II A, reaching an impressive energy per block saving (>75%).
2) Propose a multichannel and a multifrequency support, as discussed in Section II-B, to fully exploit the potentials of the proposed tunable tap approach;
3) Demonstrate in Section III how our FPGA HEVC interpolator outperforms the latest state of the art solutions, by processing UHD sequences at 60 frames/s while consuming 70% less energy per pixel.

2. Reconfigurable Approximate Interpolation

Implementation and Assessment

In recent video codecs, motion compensation is used to exploit temporal redundancy in transmitted video sequences. To compensate motion vectors with fractional values, a block is predicted by interpolating the reference block. In the HEVC standard, each luma or chroma interpolation is performed by two separable 1-D interpolation filters for the horizontal and vertical directions. These filters are carried-out with *N* taps finite impulse response filters. Most of the energy, as mentioned in [6], is consumed by these filters. To decrease the filtering complexity, legacy filters can be replaced by approximated ones with a reduced number of taps. For luminance interpolation filters, the standard filter length *N* is equal to 7 or 8; while, the approximated solution implements 7, 5, and 3 taps. For chrominance, *N* is 4 for the legacy version and 3 and 2 are considered for the approximated one. Global decoder energy is reduced by up to 20% for a software implementation on ARM and Intel platforms [6]. In the worst case, the quality degradation measured with the PSNR is around 2 dB. To avoid quality drift between reference images, the approximated filters are not applied on I-frames. The work in [12] demonstrated that such choice results in an acceptable quality drift, since the degradation due to approximated filters on reference frames is very limited, and, nevertheless, in substantial energy savings. In addition, subjective tests confirm that, even in worst cases, the degradation of the perceptual quality is considered negligible by the majority of testers [13].

A. Legacy Hardware Implementation

A hardware implementation of the HEVC motion compensation stage is desirable to reduce computational costs. Hardware acceleration relieves the host processor from an onerous interpolation workload, exploiting full parallelism and, in turn, leading to energy savings. In [9], different hardware configurations capable of performing HEVC 2-D interpolation have been presented. They range from minimal resources to maximum throughput. In this letter, aiming at energy-awareness, we opt for the baseline architecture that, according to the reported results, is capable of offering the lowest energy consumption. As depicted in Fig. 1, the architecture for an *N* taps filter is composed of

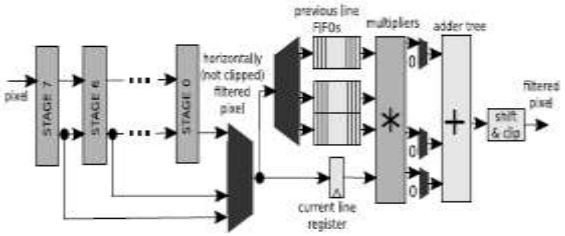

Figure 2: 2-D coarse-grained reconfigurable approximate interpolator.

a pipelined horizontal channel, which interpolates the incoming pixel rows, whose outputs (not clipped) are stored into N − 1 first-in first-out (FIFO) memories. Then, N parallel multipliers compute the product between coefficients and horizontally interpolated data (current and previous N − 1 rows in the FIFOs). An adder tree completes the vertical interpolation, prior to finalize the process with right shift and clipping phases. The *luma* baseline implementation has 8 taps and the *chroma* one has 4 taps. Hereafter, we qualify this architecture as legacy, as opposed to *the reconfigurable approximate* interpolators discussed later.

It has severely optimized the architecture presented in particular, we added pipeline stages to increase operating frequency and a systolic digital signal processor (DSP) block cascade connection similar to [14]. Table I reports a comparison between our legacy designs and those of [9]. Interpolators, as in [9], have been implemented on a Xilinx Zynq-7000 XC7Z020 SoC, with an Artix-7 28 nm FPGA. Power estimations (dP) have been carried out with Vivado Power Analysis, back-annotating the real switching activity retrieved from post-synthesis simulations running at 200 MHz. Energy per block (dE) is derived as dP ∗ Tb, where Tb is the block interpolation time. In this letter, resource demand is drastically reduced (25% and 44%) less look-up tables and flip-flops) and operating frequency is increased by 7%, resulting in an impressive energy per block saving (more than 75%). Block random access memory (BRAM)

and DSP numbers are omitted, no changes from [9] have to be reported. Throughput is one interpolated pixel per clock cycle.

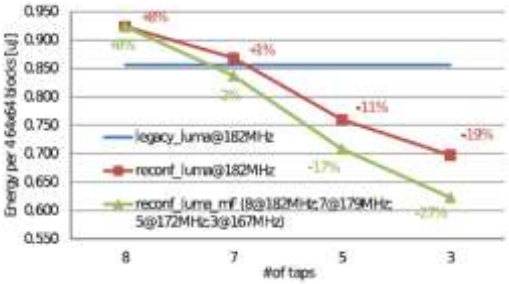

Figure 3: Four channels Luma: energy versus quality tradeoff.

B. Approximate Hardware Interpolators

Approximation is performed by coarse-grained reconfigurable interpolators for luma (*reconf_luma*) and chroma (*reconf_chroma*) starting from the baseline architecture, as depicted in Fig. 2. The idea is to provide runtime adaptively among different data paths, avoiding complete or partial reconfiguration and getting rid of any extra energy overhead. The coarse-grained approach requires instantiating multiplexers to exclude certain stages from the computation upon request. Clock gating and operand isolation have been inserted to limit dynamic power consumption, respectively in the unused flipflops and logic of the excluded stage. As in [6] and [9], *reconf_luma* performs 8/7 (legacy), 5 and 3-tap filtering; while *reconf_chroma* provides 4 (legacy), 3 and 2-tap.

The reconfigurable designs with respect to legacy, non-reconfigurable ones. Reconfiguration implies a severe overhead in terms of logic cells and registers, while BRAMs and DSPs remain unaltered. Operating frequency is affected: 6% drop in *luma* and 12% in *chroma*. Nevertheless, power and energy results motivate such overhead: except in the maximum tap configurations and approximate designs consume less energy than legacy ones. This achievement is clear in Figs. 3 and 4 where *reconf_luma* and *reconf_chroma* are able to provide a tradeoff between system consumption (energy per block) and interpolation quality (# of taps). These results confirm the trend in [9], but Figs. 3 and 4 refer to four parallel interpolation channels, which is a more realistic scenario and limits eventual rounding errors (due to small estimation numbers).

Operating frequency decreases to 182 MHz for *luma* and 193 MHz for *chroma*. Figs. 3 and 4 present the behavior of a third couple of designs, based on a multifrequency approximate solution. Considering a positive side effect results from taps reduction: the shorter

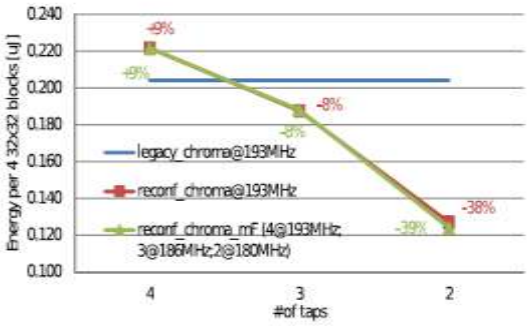

Figure 4: Four channels chroma: energy versus quality tradeoff.

the filter and the faster each block is processed. As a consequence, a higher throughput is obtained on a fixed time-frame. The *# of interpolated pixels*, in the considered time-frame, increases by up to 9% going from 8 to 3 taps in the *luma* case and by up to 7% going from 4 to 2 taps in the *chroma* case. With respect to [9], two additional designs are assessed: 1) *reconf_luma_mF* and 2) *reconf_chroma_mF*.

Their frequency is reduced when nonlegacy interpolations are executed, while guaranteeing at least the same legacy interpolators throughput. The resulting *reconf_luma_mF* and *reconf_chroma_mF* designs save, respectively, up to 29% and 40% of dynamic power, even though their execution time is larger. This is why improvements on the energy tradeoff are clearly visible only in Fig. 3.

3. Comparison with the State of the Art

We limit our discussion to recent FPGA approaches capable of supporting at least FHD (1920 × 1080 pixels). Here follows an excursus on the considered implementations, while Section III-A reports performance comparison. Alfonso et al. [15] proposed a QFHD (3840 × 2160) interpolator working at 30 frames/s, managing 8×8 blocks and not supporting *chroma*. In [16], reconfiguration is adopted to provide different fractional shifts, minimizing resources: *chroma* is not supported and basic block size is 8 ×8. Generally speaking, FPGA implementations are meant to provide high throughput, but are not energy efficient by nature. The most common way to design low power filters is to adopt multiplier-less solutions, where multiplications are substituted by shifters and adder trees (filter coefficients are fixed a priori). Kalali and Hamzaoglu [17] designed a multiplier-less circuit with a reduced number and size of the adders and minimizing the adder tree depth. They reach QFHD (3840 × 2160) resolution at 30 frames/s with a limited energy consumption. Diniz et al. [18] exploited data dependency and develop a partially reconfigurable multiplier-less design, customized on the current group of pictures to be decoded. Customization minimizes resources, limiting the power of overestimated logic. Alongside all the previous highly manually optimized solutions, Ghani et al. [19] adopted high level synthesis to speed up a QFHD real-time interpolation accelerator deployment. Our full scale demonstrator does not leverage on a multiplier-less approach to save power. We implement other direct energy reduction techniques (i.e., operand isolation and clock gating) and we leverage on computation approximation and multifrequency.

A. Performance Comparison

Before comparing the proposed work with state of the art, a dimensioning step is mandatory. To process UHD resolution video sequences at 60 frames/s and with 4:2:2 chroma subsampling, the interpolator has to provide 500 and 250 Mpixels/s, respectively, for *luma* and for *chroma*. We support a throughput of one pixel per cycle. Clocked at 180 MHz, three *luma* and four *chroma* in parallel meet the required throughput constraints. In terms of energy versus quality, three models are used: 1) high, *H*, where the legacy filters (8/7-tap *luma* and 4- tap *chroma*) is executed; 2) medium, *M*, with 5-tap *luma* and 3-tap *chroma*; and 3) low, *L*, with 3-tap *luma* and 2-tap *chroma*. Comparison of the proposed design to related works is reported in Table III. Besides the Zynq-7000 FPGA, for a fair comparison we provide also results on a Virtex-5. Our design, that is the only non multiplier-less solution and uses several row FIFOs, employs the largest amount of BRAMs and multipliers, but consumes the lowest amount of energy per interpolated pixel, even if the worst technology point is considered. On top of that, multifrequency support guarantees further energy reduction, when *M* and *L* approximated filters are
adopted.

In resolution and power, with the latest literature works, with the further benefit of providing dynamic quality versus energy tradeoff. In comparison with the lowest power consuming literature design [18], it saves 70% energy per pixel while running in *H* mode and up to 78% (82% with multifrequency) in *L* mode.

4. Conclusion

HEVC is the last and most efficient video codec standard defined so far. It requires several computational intensive elaboration steps, among which fractional pixel interpolation is the topmost one. Boosting HEVC interpolation is challenging. In this letter, we proposed and assessed an FPGA interpolator for HEVC able to process UHD sequences at 60 frames/s. Our implementation exploits approximate computing, combined with coarse-grained reconfiguration, to provide energy versus quality tradeoffs. The proposed reconfigurable, multichannel, and multifrequency, approximate computing implementation beats the most energy efficient FPGA implementations in literature, consuming from 70% to 82% less energy per pixel.

References

[1] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards— Including high efficiency video coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[2] L. Renganarayana, V. Srinivasan, R. Nair, and D. Prener, "Programming with relaxed synchronization," in *Proc. ACM RACES Workshop*, Tucson, AZ, USA, 2012, pp. 41–50.

[3] D. Palomino, M. Shafique, A. Susin, and J. Henkel, "Thermal optimization using adaptive approximate computing for video coding," in *Proc. DATE Conf.*, Dresden, Germany, 2016, pp. 1207–1212.

[4] T. Hilaire, D. Menard, and O. Sentieys, "Bit accurate roundoff noise analysis of fixed-point linear controllers," in *Proc. IEEE CACSD Conf.*, San Antonio, TX, USA, 2008, pp. 607–612.

[5] M. Shafique, R. Hafiz, S. Rehman, W. El-Harouni, and J. Henkel, "Invited: Cross-layer approximate computing: From logic to architectures," in *Proc. ACM/EDAC/IEEE DAC Conf.*, Austin, TX, USA, 2016, pp. 1–6.

[6] E. Nogues, D. Menard, and M. Pelcat, "Algorithmic-level approximate computing applied to energy efficient HEVC decoding," *IEEE Trans. Emerg. Topics Comput.*, to be published.

[7] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012.

[8] J. A. Gomez-Pulido, P. J. Cordeiro, and P. A. Assunção, "Performance, power and scalability analysis of HEVC interpolation filter using FPGAs," in *Proc. IEEE EUROCON Conf.*, 2015, pp. 1–6.

[9] F. Palumbo *et al.*, "Runtime energy versus quality tuning in motion compensation filters for HEVC," in *Proc. PDeS Conf.*, Brno, Czech Republic, 2016, pp. 145–152.

[10] S. Khan and E. Casseau, "High-performance motion estimation operator using multimedia oriented subword parallelism," *J. Commun. Comput.*, vol. 9, no. 1, pp. 1–14, 2012.

[11] L. Liu *et al.*, "An implementation of multiple-standard video decoder on a mixed-grained reconfigurable computing platform," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 5, pp. 1285–1295, 2016.

[12] E. Nogues, E. Raffin, M. Pelcat, and D. Menard, "A modified HEVC decoder for low power decoding," in *Proc. ACM CF Conf.*, 2015, p. 60.

[13] N. Sidaty *et al.*, "Reducing computational complexity in HEVC decoder for mobile energy saving," in *Proc. EUSIPCO Conf.*, 2017.

[14] *DSP: Designing for Optimal Results*, Xilinx, San Jose, CA, USA, 2005.

[15] V. Afonso, H. Maich, L. Agostini, and D. Franco, "Low cost and high throughput FME interpolation for the HEVC emerging video coding standard," in *Proc. IEEE LASCAS Conf.*, Cusco, Peru, 2013, pp. 1–4.

[16] E. Kalali, Y. Adibelli, and I. Hamzaoglu, "A reconfigurable HEVC sub-pixel interpolation hardware," in *Proc. IEEE ICCE Conf.*, Berlin, Germany, 2013, pp. 125–128.

[17] E. Kalali and I. Hamzaoglu, "A low energy HEVC sub-pixel interpolation hardware," in *Proc. IEEE ICIP Conf.*, Paris, France, 2014, pp. 1218–1222.

[18] C. M. Diniz, M. Shafique, S. Bampi, and J. Henkel, "A reconfigurable hardware architecture for fractional pixel interpolation in high efficiency video coding," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 2, pp. 238–251, Feb. 2015.

[19] F. A. Ghani, E. Kalali, and I. Hamzaoglu, "FPGA implementations of HEVC sub-pixel interpolation using high-level synthesis," in *Proc. IEEE DTIS Conf.*, Istanbul, Turkey, 2016, pp. 1–4.